

Foile de stil în paginile Web

Prof.dr. Floarea NASTASE
Catedra de Informatica Economica, A.S.E. Bucuresti

Foile de stil reprezinta un câstig major al proiectantilor de pagini Web, oferind posibilitati suplimentare pentru prezentarea documentelor. Prin separarea stilului de prezentare a documentului de continutul sau, se simplifica realizarea si întretinerea site-urilor. Limbajul normalizat al foilor de stil CSS, deja utilizat în HTML, va putea fi utilizat la concurenta sau va fi înlocuit de XSL.

Cuvinte cheie: CSS, XSLT, HTML, XML, XPath.

C e sunt foile de stil?

Foile de stil descriu modul în care documentele vor fi prezentate. W3C a promovat utilizarea foilor de stil pe Web printr-un Consorțiu fondat în 1994. Activitatea s-a concretizat prin realizarea celor doua recomandari pentru foile de stil: **CSS1**, **CSS2** (Cascading Style Sheets, level 1 si level 2) care sunt, în mare masura, implementate în browser-e. Noua specificatie CSS Mobile Profile 1.0 definește un subset al CSS2 adaptat necesitatilor si restrictiilor impuse de dispozitivele mobile. De asemenea, activitatea grupului W3C include si dezvoltarea foilor de stil **XSL** (Extensible Style Language).

CSS2 este un limbaj al foilor de stil care permite programatorilor si utilizatorilor sa ataseze stiluri documentelor structurate, cum ar fi paginile HTML sau aplicatiile XML. CSS2 ofera foi de stil pentru diverse media, astfel se autorizeaza o mai buna prezentare a documentelor prin navigatoare vizuale, dispozitive auditive, imprimante, dispozitive Braille etc.

Cu toate ca W3C a lansat dezvoltarea XSL ca o completare la CSS, au aparut o serie de confuzii. De ce se dezvolta un al doilea limbaj pentru foile de stil când implementarea pentru primul nu s-a terminat? Daca se face o scurta comparatie între CSS2 si XSL se poate gasi raspunsul:

Criteriul de comparatie	CSS2	XSL
Posibilitatea de utilizare cu HTML	da	nu
Posibilitatea de utilizare cu XML	da	da
Utilizarea sintaxei XML	nu	da
Transformarea structurii documentelor XML	nu	da
Transformarea continutului documentelor XML	nu	da
Sortarea, cautarea, filtrarea datelor XML	nu	da
Suportat de navigatoarele obisnuite (Internet Explorer 5.0 si Netscape Navigator 6.)	da	partial

Unicul avantaj al CSS este acela ca poate fi utilizat pentru prezentarea documentelor HTML. XSL, pe de alta parte, are capacitatea sa transforme documentele. De exemplu, XSL poate fi utilizat pentru a transforma un document XML într-un document HTML/CSS pe server-ul Web. În acest fel, cele doua limbaje se completeaza

reciproc si pot fi utilizate împreuna. Ambele limbaje pot fi utilizate pentru prezentarea documentelor XML.

Structura unei foi de stil CSS

O foaie de sil CSS contine un set de *reguli* si, optional, *comentarii*. De exemplu, pentru a specifica o anumita culoare (rosu)

textului delimitat de un marcator (H1), trebuie sa existe o *regula de stil* CSS de forma:

```
H1 { color: red }
```

Fiecare *regula de stil* CSS este compusa din:

- *selector* (în exemplu: 'H1') - indica elementul sau elementele asupra carora actioneaza regula;
- *blocul pentru declaratii* ('color: red') - descrie conditiile de redare fizica a ele-

```
Selector { propr1:val1; propr2: val2;...}
```

Astfel, regulile urmatoare:

```
H1 { font-weight: bold }
H1 { font-size: 12pt }
H1 { line-height: 14pt }
H1 { font-family: Helvetica }
H1 { font-variant: normal }
H1 { font-style: normal }
```

sunt echivalente cu:

```
H1 {
font-weight: bold;
font-size: 12pt;
line-height: 14pt;
font-family: Helvetica;
font-variant: normal;
font-style: normal
}
```

Comentariile CSS furnizeaza unele informatii celor care studiaza codul sursa si sunt ignorate de navigator. Un comentariu CSS începe cu "/*" si se încheie cu "*/", de exemplu:

```
P
{
/* Acest paragraf este centrat */
text-align: center;
/* Culoarea utilizata pentru text este rosu */
color: red;
}
```

Specificatiile HTML 4.0 definesc modul în care regulile de stil pot fi utilizate pentru documentele HTML: fie în documentul HTML sau într-un fisier extern.

Stiluri CCS în paginile HTML

Pentru ca foile de stil sa poata influenta prezentarea, navigatorul trebuie sa aiba cu-

mentului specificat, fiind delimitat de acolade.

La rândul sau, declaratia include doua parti: *proprietatea* ('color') si *valoarea* ('red'). Numarul de declaratii (*proprietate-valorie*) incluse într-o regula nu este limitat. Separatorul între declaratii este punct-si-virgula (;). Sintaxa generala a unei reguli de stil se prezinta astfel:

nostinta de existenta lor. În exemplul ur-mator se prezinta un cod sursa HTML, care face apel la foi de stil:

```
<HTML>
<HEAD>
<TITLE>Exemplu 1</TITLE>
<LINK REL=STYLESHEET TYPE="text/css"
  HREF="http://style.com/cool" TITLE="Cool">
<STYLE TYPE="text/css">
  @import url(http://style.com/basic);
  H1 { color: blue }
</STYLE>
</HEAD>
<BODY>
<H1>Titlul este albastru</H1>
<P STYLE="color: red">Acest paragraf este
rosu
</BODY>
</HTML>
```

Se remarca prezenta a patru moduri pentru combinarea foilor de stil cu pagina HTML:

- utilizarea elementului 'LINK' pentru a face legatura catre o foaie de stil externa;
- declararea elementului 'STYLE' în cadrul elementului 'HEAD';
- importarea unei foi de stil, se utilizeaza notatia CSS '@import';
- asocierea atributului 'STYLE' unui element din corpul documentului ('BODY').

Stiluri CSS în paginile XML

Foile de stil pot fi asociate cu un document XML prin utilizarea unei instructiuni de prelucrare a carei tinta este xml-stylesheet.

Acesta instructiune de prelucrare actioneaza în acelasi mod ca tag-ul <LINK REL="stylesheet"> din HTML 4.0. Instructiunea de prelucrare xml-stylesheet este inclusa numai în prologul unui document XML.

Sintaxa generala pentru apelarea unei foi de stil CSS dintr-un document XML printr-o instructiune de prelucrare este:

```
<?xml-stylesheet href="nume_fisier.css" type="text/css" ?>
```

XML depinde mai mult de foile de stil decât HTML. În continuare este prezentat un document XML foarte simplu, caruia i s-a asociat o foaie de stil CSS:

```
<?xml version="1.0" encoding="ISO8859-2" ?>
<?xml-stylesheet href="ex1_xml.css" type="text/css"?>
<BIBLIO>
  <TITLUL> Bibliografie </TITLUL>
  <CARTEA ISBN="973-590-132-3" LIMBA="ro" SUBIECT="Web">
    <AUTOR>Pavel Nastase</AUTOR>
    <AUTOR>Floarea Nastase</AUTOR>
    <TITLUC>Internet - WWW, JavaScript, HTML, JAVA</TITLUC>
    <EDITURA> Economica, Bucuresti</EDITURA>
    <DATAPUB>1998</DATAPUB>
  </CARTEA>
  <CARTEA ISBN="9782212090529" LIMBA="fr" SUBIECT="general">
    <AUTOR> Alain Michard </AUTOR>
    <TITLUC>XML, Langage et Applications</TITLUC>
    <EDITURA> Eyrolles, Paris</EDITURA>
    <DATAPUB>1998</DATAPUB>
  </CARTEA>
  <CARTEA ISBN="973-590-253-2" LIMBA="ro" SUBIECT="general">
    <AUTOR> Floarea Nastase</AUTOR>
    <TITLUC>Arhitectura rețelelor de calculatoare</TITLUC>
    <EDITURA> Economica, Bucuresti</EDITURA>
    <DATAPUB>1999</DATAPUB>
  </CARTEA>
  <CARTEA ISBN="9782840825685" LIMBA="fr" SUBIECT="aplicatii">
    <AUTOR> William J. Pardi </AUTOR>
    <TITLUC>XML en Action</TITLUC>
    <EDITURA> Microsoft Press, Paris</EDITURA>
    <DATAPUB>1999</DATAPUB>
  </CARTEA>
</BIBLIO>
```

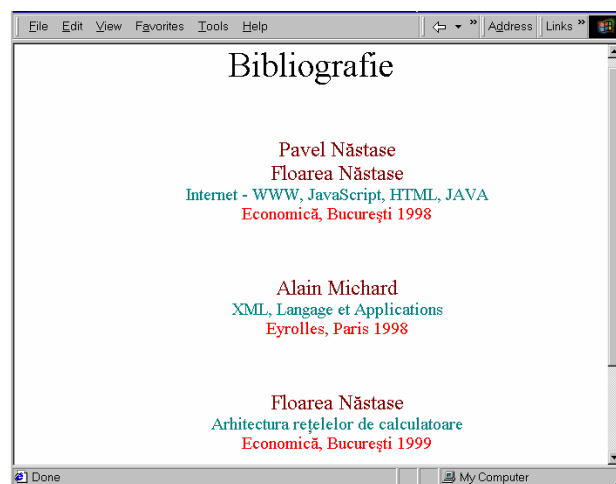


Fig.1. Rezultatul afisarii cu navigatorul IE 5.0

Pentru a afisa acest fragment XML într-un anumit model de document, trebuie mai întâi declarate regulile de stil. Adaugând mai multe linii în foaia de stil se va îmbunătăți modul de prezentare a documentului. Pentru a obține imaginea din figura 1 conținutul fișierului ex1_xml.css este:

```
TITLUL{ text-align: center;
font-size: 200%;
color: black;
text-shadow: black 0px 0px 5px;
display: block
}
CARTEA {text-align: center;
margin-top: .5in;
margin-bottom: .5in;
margin-left: 1.5in;
margin-right: 1in;
color: red;
background-color: white;
font-family: Verdanaserif;
font-size: 100%;
display: block
}
AUTOR { font-size: 120%;
font-family: Ariansans-serif;
color: maroon;
display: block
}
TITLUC {font-size: 100%;
font-family: Ariansans-serif;
color: teal;
display: block
}
```

Foile de stil XSL

XSL este un limbaj pentru descrierea stilului de prezentare a documentelor XML într-un mod mai evoluat decât CSS. O foaie de stil XSL este un document XML bine-format. Limbajul de stil extensibil XSL se utilizează atât pentru prelucrarea datelor XML cât și pentru descrierea formatului informației conținute în documentele XML.

În realitate, XSL include trei limbaje:

- **XSLT** (XSL Transformations) - limbaj de **transformare XML**;
- **XPath** (XML Path Language) - limbaj pentru **adresarea fragmentelor XML**;
- XSL Formatting Objects - limbaj care definește modul de afișare a documentului XML.

Atât limbajul de transformare cât și limbajul de formatare constituie aplicații XML. Partea de transformare și partea de formatare a XSL pot funcționa independent. De exemplu, limbajul de transformare poate converti un document XML într-un document HTML bine-format, ignorând în totalitate obiectele de formatare XSL. Limbajul de transformare reunește elementele care permit definirea regulilor pentru transformarea unui document XML în alt format de document.

În procesul de transformare, XSLT utilizează XPath pentru identificarea fragmentelor din documentul sursă care corespund unuia sau mai multor șabloane (*templates*) definite. Când o corespondență este găsită, XSLT va transforma fragmentul respectiv, al documentului sursă, într-un fragment al documentului rezultat.

Fiecare foaie de stil XSL descrie regulile pentru prezentarea unei clase de documente sursă XML. Procesul de prezentare include două componente:

- generarea arborelui rezultat, având ca intrare arborele documentului sursă;
- interpretarea arborelui rezultat, pentru a se obține un format al ieșirii care să poată fi afișat, imprimat, audiat etc.

XSL poate fi utilizat pentru a transforma un document XML într-un alt tip de document care va fi recunoscut de un navigator, cum ar fi formatul HTML (fig. 2). XSL nu poate cere arborelui rezultat să utilizeze vocabularul de formatare, acesta poate fi utilizat pentru transformarea generală XML.

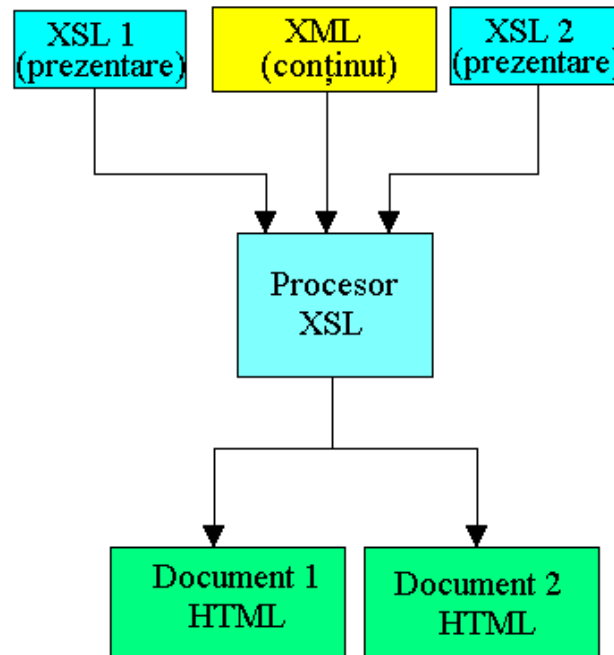


Fig.1. Generarea documentelor HTML

Transformarea unui document XML prin XSLT

O foaie de stil XSLT conține un set de reguli *template*. O regulă *template* definește un *pattern*, care permite să se știe care fragment al arborelui sursă se aplică regula, iar un *template* va fi instantiat pentru a forma o parte din arborele rezultat. Aceasta facilitează unei foi de stil XSLT să fie aplicabilă unei largi categorii de documente care au structurile arborilor sursă similare.

Template-ul poate conține elemente prin care se specifică structura unor elemente rezultat de tip caracter (tag-uri, date noi și date recuperate din documentul XML original) și elemente instrucțiuni pentru generarea fragmentelor din arborele rezultat. Când un *template* este instantiat, fiecare instrucțiune este executată și înlocuită prin fragmentul din arborele rezultat generat. Instrucțiunile pot selecta și prelucra elementele descendente. Prelucrarea elementelor descendente creează un fragment al arborelui rezultat prin găsirea *regulii template* aplicabile și instantierea ei. De notat că, elementele sunt prelucrate numai când ele sunt selectate prin execuția unei instrucțiuni. Arborele rezultat este construit

prin găsirea regulii *template* pentru nodul rădăcină și instantierea *template*-ului său.

În procesul de găsire a regulii *template* aplicabile, mai mult decât o regulă *template* poate avea un *pattern* care se potrivește cu un element al documentului sursă. Totuși, numai o regulă *template* poate fi aplicată.

Instantierea unui *template* se face prin raportarea la un *nod curent* și o *listă curentă a nodurilor*. Nodul curent este totdeauna membru al listei curente de noduri. Multe din operațiile XSLT sunt relative la nodul curent. Numai câteva instrucțiuni permit schimbarea listei curente a nodurilor sau a nodului curent. Pe durata instantierii uneia din aceste instrucțiuni, lista curentă a nodurilor este modificată și se formează o nouă listă curentă a nodurilor ale căror elemente devin, pe rând, nod curent. La sfârșitul acestui proces, nodul curent și lista curentă a nodurilor redevine ceea ce era înainte de instantierea instrucțiunii.

XSLT utilizează limbajul expresiilor definit prin XPath pentru o diversitate de situații:

- selectarea nodului de prelucrare;
- specificarea condițiilor pentru diferitele moduri de prelucrare ale unui nod;

- generarea textului care se va insera în arborele rezultat.

Expresiile intervin ca valori ale anumitor atribute ale elementelor XSLT si ca *template*-uri pentru valorile atributelor, când sunt amplasate între acolade.

Structura unei foi de stil XSLT

O foaie de stil XSLT într-un format simplificat, declarata implicit, este un fisier XML care contine numai un singur *template*

pentru nodul radacina. Foaia de stil poate contine un simplu element literal rezultat, reprezentând rezultatul efectiv al transformarii (figura 3). Arborele foi de stil formeaza *template*-ul utilizat prin procesorul XSLT pentru generarea arborelui rezultat. Daca nu sunt instructiuni XSLT, arborele foi de stil devine arborele rezultat. Daca sunt prezente instructiuni, procesorul le înlocuieste cu rezultatele executiei lor.

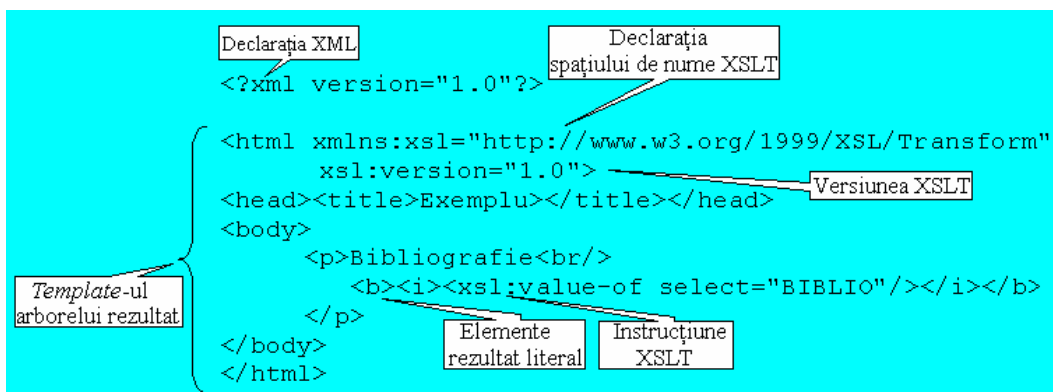


Fig.2. Foaie de stil declarata implicit

Declaratia XML este interpretata de procesorul XML. Restul fisierului este considerat *template*-ul arborelui rezultat pentru o regula implicita a radacinii documentului, descriind rezultatul transformarii. În acest caz, elementul document este denumit "html".

Vom considera documentul XML prezentat în paragraful "Stiluri CSS în paginile XML", caruia îi vom asocia o foaie de stil XSLT prin instructiunea de prelucrare:

```
<?xml-stylesheet type="text/xsl" href="ex1.xsl" ?>
```

Continutul fisierului ex1.xsl este:

```
<?xml version="1.0" ?>
<HTML xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<BODY STYLE="font-family:Arial, helvetica, sans-serif;
font-size:12pt;
background-color:#EEEECC">
<xsl:for-each select="BIBLIO/CARTEA">
```

```
<DIV STYLE="background-color:teal;
color:white;
padding:4px">
<SPAN STYLE="font-weight:bold; color:white">
<xsl:for-each select="AUTOR">
<xsl:value-of select="." /> <br/>
</xsl:for-each>
</SPAN>
</DIV>
<DIV STYLE="background-color:#AACCCC;
color:white;
font-size:14pt;
padding:4px">
<xsl:value-of select="TITLUC" />
</DIV>
<DIV STYLE="margin-left:20px;
margin-bottom:1em;
font-size:10pt">
Editura <xsl:value-of select="EDITURA" />,
<xsl:value-of select="DATAPUB" />
</DIV>
</xsl:for-each>
</BODY>
</HTML>
```

Rezultatul vizualizarii documentului XML cu navigatorul Internet Explorer 5.0 este reprezentat în figura 4.

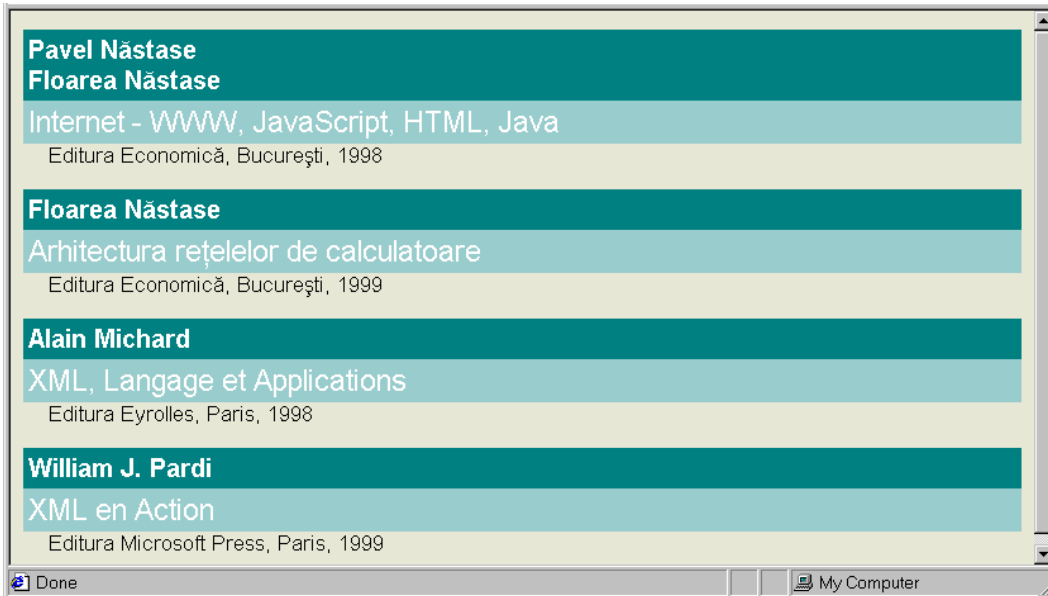


Fig.3. Vizualizarea unui document XML (foaie de stil implicita)

În mod frecvent se utilizeaza foaia de stil XSLT delimitata printr-un element care contine toate specificatiile, declarata explicit (figura 5). Acest element poate fi instructiunea XSLT stylesheet sau transform care trebuie sa includa prefixul asociat

URI-lui spatiului de nume XSLT. Instructiunea stylesheet este un element radacina pentru o foaie de stil, dar în anumite situatii poate fi continuta într-un document XML.

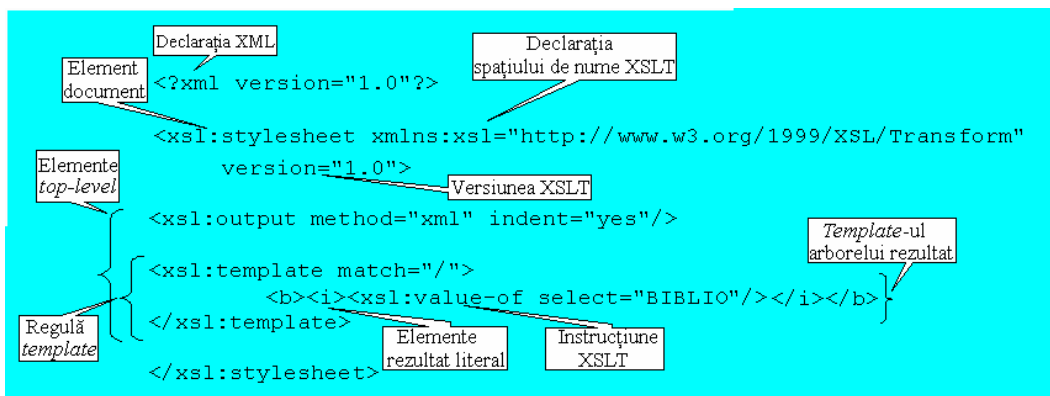


Fig.4. Foaie de stil explicita

O foaie de stil declarata implicit poate fi echivalenta cu o foaie de stil care utilizeaza un element `xsl:stylesheet` incluzând o regula *template* ce corespunde elementului radacina si continând elementul literal rezultat.

Orice foaie de stil XSLT este un document care utilizeaza sintaxa XML si poate începe cu o declaratie în acest sens:

```
<?xml version="1.0"?>
```

Toate foile de stil XSLT se autodeclara ca fiind foi de stil. În plus, tag-urile utilizeaza

conceptul spatiului de nume, astfel încât fiecare element XSLT contine declaratia spatiului de nume "xsl:".

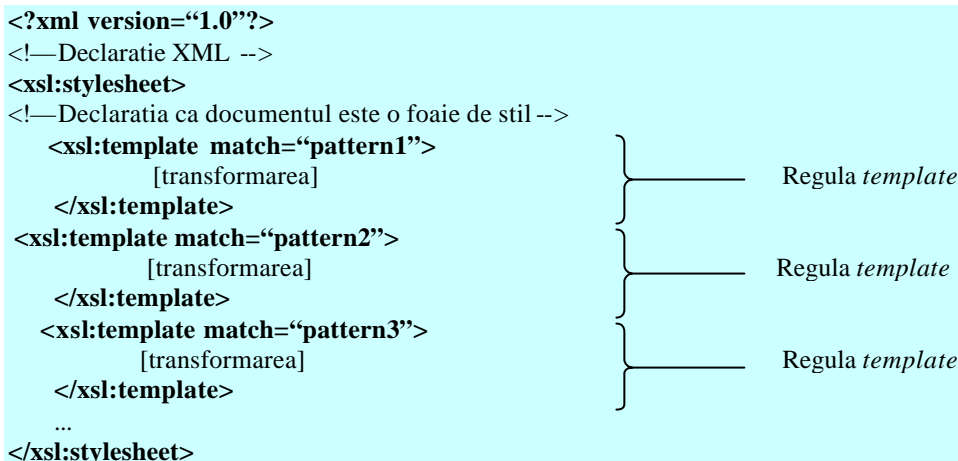
Elementul de baza al unei foi de stil XSLT, instructiunea `xsl:stylesheet`, are sintaxa:

```
<xsl:stylesheet
  id = id
  extension-element-prefixes = tokens
  exclude-result-prefixes = tokens
  version = numar_versiune>
  <!-- Continut (xsl:import*, elemente top-level)-->
</xsl:stylesheet>
```

Instructiunea `xsl:transform` este acceptata ca un element sinonim pentru `xsl:stylesheet`.

Elementul `xsl:stylesheet` se regaseste în fiecare foaie de stil XSLT si reprezinta no-

dul radacina al acesteia. El informeaza procesorul XSLT ca acest document este o foaie de stil si gazduieste *template*-uri. Structura simplificata a unei foi de stil este:



Fiecare regula *template* este reprezentata print-un element `xsl:template`. *Pattern*-ul unei reguli este valoarea atributului `match` al elementului `xsl:template`. Sablonul pentru iesire este continutul elementului `xsl:template`. Toate instructiunile dintr-un *template* prin care se realizeaza anumite operatii, cum ar fi selectarea elementelor din arborele de intrare care vor fi incluse în arborele de iesire, sunt elemente XSLT. Acestea sunt identificate prin prefixul `xsl:` în numele elementului. Elementele care nu au un prefix `xsl:` se regasesc în arborele rezultat.

Elementul radacina `xsl:stylesheet` detine doua attribute obligatorii, `version` si `xmlns:xsl`, fiecare dintre ele trebuie sa aiba exact valorile specificate (1.0 pentru `version` si `http://www.w3.org/1999/XSL/Transform` pentru `xmlns:xsl`).

Elementul `xsl:stylesheet` poate contine urmatoarele tipuri de elemente:

- `xsl:import`
- `xsl:include`
- `xsl:strip-space`
- `xsl:preserve-space`
- `xsl:output`
- `xsl:key`
- `xsl:namespace-alias`

- `xsl:attribute-set`
- `xsl:variable`
- `xsl:param`
- `xsl:template`

Un element fiu al elementului `xsl:stylesheet` este denumit element de *nivel superior* (*top-level*). Instructiunile *top-level* sunt analizate si/sau executate de procesorul XSLT înainte începerii pelucrării informatiei din arborele sursa.

În mod normal, un element de nivel superior este continut într-o foie de stil XSLT o singura data, de mai multe ori sau niciodata. Structura unei foi de stil, la modul foarte general este:

```

<xsl:stylesheet
version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:import href="..."/>
  <xsl:include href="..."/>
  <xsl:strip-space elements="..."/>
  <xsl:preserve-space elements="..."/>
  <xsl:output method="..."/>
  <xsl:key name="..." match="..." use="..."/>
  <xsl:namespace-alias stylesheet-prefix="..."
result-prefix="..."/>
  <xsl:attribute-set name="...">
  ...
</xsl:attribute-set>
  <xsl:variable name="...">...</xsl:variable>
    
```



```

<xsl:param name="...">...</xsl:param>
<xsl:template match="...">
...
</xsl:template>
<xsl:template name="...">
...
</xsl:template>
</xsl:stylesheet>

```

Punctele de suspensie (...) indica pozițiile în care valorile atributelor sau conținutul elementelor au fost omise. În aceasta foaie de stil fiecare element autorizat este regăsit o singură dată. Ordinea în care subelementele elementului `xsl:stylesheet` apar nu are importanță, exceptând elementele `xsl:import`. Utilizatorii au libertatea de a ordona elementele după preferințele lor. Filozofia XML constă într-o separare a datelor/documentelor (fișierele XML propriu-zis) de prelucrări/prezentări. Un document dat va fi, după crearea sa, marcat *unic* în funcție de conținutul său și *independent* de redarea sa ulterioară - ecran, hârtie, terminal Braille, sintetizor vocal - ca

și de alte prelucrări automate care se pot aplica.

Această independență va conferi:

- o foarte mare *interoperabilitate* - același document va putea fi afișat pe Web și/sau va fi tipărit pe suport hârtie, va putea fi înregistrat într-un SGBD etc.;
- o foarte mare *durabilitate/reutilizabilitate* - documentul nu va trebui să fie supus evoluției tehnologiei informației, putând fi încorporat, fără dificultate, în documente de natură foarte diferite.

Bibliografie

- **XML** - <http://www.w3.org/TR/1998/REC-xml-19980210>
- **XML Names** - <http://www.w3.org/TR/REC-xml-names>
- **XPath** - <http://www.w3.org/TR/xPath>
- **CSS2** - <http://www.w3.org/TR/1998/REC-CSS2-19980512>